UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/689,126	10/20/2003	John B. Condon	BLD920030025US1	2905

50441 7590 11/24/2008

DUFT BORNSEN & FISHMAN, LLP
1526 SPRUCE STREET
SUITE 302
BOULDER, CO 80302

EXAMINER	
DICKERSON, CHAD S	

ART UNIT	PAPER NUMBER
2625	

MAIL DATE	DELIVERY MODE
11/24/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/689,126	CONDON ET AL.	
	Examiner	Art Unit	
	CHAD DICKERSON	2625	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *31 July 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-4,9-13,18-21 and 25* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-4, 9-13, 18-21 and 25* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *10/20/2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Arguments

1. Applicant's arguments filed 7/31/2008 have been fully considered but they are not persuasive. In the response filed, the Applicant made several assertions. First, the reference of Salgado does not perform the feature of having print jobs parsed into a plurality of work units and the Salgado reference also does not perform the queuing the plurality of work units in the manner recited in claim 1. Next, the Applicant asserts that the Wood reference not only does not teach processing work units independently, but it also is improperly combined with the reference of Motamed. The Examiner would like to respond to these assertions below.

Regarding the argument of the Wood reference not disclosing processing of the work units independently, the Examiner disagrees with this assertion. The Examiner viewed in Applicant's specification the feature regarding the independence of the work unit. As understood to the Examiner, in paragraph [0022], the independence of the work unit is interpreted as the work unit containing all the information within the work unit for it to be processed in the system. When looking at the Wood reference, this feature is performed along with the features that were previously suggested to be taught by Motamed. The Examiner will explain how the reference of Wood alone performs the features previously rejected by Motamed and therefore, the reference of Motamed is no longer needed in rejecting the independent claims.

The features of that were previously rejected by Motamed consists of "*wherein each work unit may be processed independent of all other work units and wherein the*

plurality of work units are parsed from a single job and wherein the processing of each work unit is independent of processing of the other work units and wherein multiple work units are processed in parallel by multiple compute nodes". Regarding the above features, the Wood reference does, in fact, perform all of the above. The Applicant quoted a section in paragraph [0034] of Wood that disclosed the order-dependence of the data files as teaching that the Wood reference does not disclose the work units as being independent of others. Before responding to the validity of this conclusion, the Examiner would like to present the background of the invention as a foundation of the argument below since the background is improved upon to create the Wood invention.

In the background, it was duly noted by Wood of the present inventions used to take a PDL datastream and parse it into multiple segments (mentioned in paragraph [0004]). This statement alone performs the claimed feature of "*wherein the plurality of work units are parsed from a single job".* Since the multiple segments are to be converted into another intermediate data stream, or another format, the segments can be considered analogous to a work unit. Wood also noted that the prior art divided the PDL data stream into independent data stream portions by multiple sub-processes. The prior art system ensured that the independence of the data stream segments were to be maintained through having translation information containing the appropriate "translation state" for each segment. Here, the segments could be processed by only depending on the system being able to process the translation information contained in the segment and not in any other segment (see paragraphs [0004]-[0006]). This statement here performs another claim feature stating "*wherein the processing of each work unit is*

independent of processing of the other work units". A further indicator of their independence was the various portions of the intermediate data stream portion being combined into a single intermediate data stream portion. Since these portions were not generated in chronological order, the portions would be generated at whatever rate the PDL interpreters could generate the information and then the system would have to then rearrange the various data in a correct chronological order to obtain the original data stream. This is another example of the portions being processed independently of each other (see paragraph [0007]).

However, when looking at the reasons for Wood's invention, it is to improve upon the invention of the prior art. Wood thought it was a problem to have different segments generated at different times and then have these segments re-aggregated in the correct chronological order. In order to improve upon the prior art, Wood decided to have different PDL processors process different segments in a manner to easily reflect the chronological order. When the Applicant mentioned how this feature proves that the segments are not independent from each other, the Examiner disagreed with this statement. In paragraph [0034], Wood simply allows for a PDL processor to create a global file to be processed on a processor that contains information that might affect subsequent segments, meaning that portions of a page processed on the same processor may have the same font and the global file is used to guarantee consistent fonts on the same page. However, the fonts could be different from portion to portion or page to page since the statement mentioned in paragraph [0029] mentions that the global data file contains PDL operations that might affect others, which may mean that it

might not affect other portions. The reasoning behind the use of the system quoted by the Applicant (see page 9 of Applicant's response) is to make sure that if PDL processor 1 contains data for pages 1-3, the processor processes the pages in chronological order from 1 to 3 in order to cut down of the complexity of the system. The complexity of the system is through taking data generated in any order and then taking complicated steps to ensure correct chronological ordering of the pages. This same example goes for 3 portions that make up a page. The global data file is used to simply tell pages 1 to 3, or different portions on a page, that they should all have 12 point font. If the system is not dealing with pages 1-3 but three portions on a single page, it would not look to appealing to have one portion with 10 point font and the other portions with 12 and 14 point font unless this is the case for the image data. Thus, the global data file is used to make sure that the font in all three portions is consistent with the original image data. Therefore, since the segments contain both a global data file and a segment data file, the processors have all that they need to process a segment independently from other segments. Although, since different segments on a processor can have different global data files as shown in figure 5B, the system again performs the feature of independently processing segments, which are analogous to work units (see paragraph [0034] and [0035]).

Regarding the last feature, the Wood reference noticed that the prior art had a short coming since it could not perform parallel processing of multiple segments of a PDL data stream. The Wood reference improved on this shortcoming by having an apparatus that can perform this feature (see paragraphs [0008]-[0013]). The different

PDL processors are considered analogous to compute nodes. With the above disclosure of the Wood reference, the feature of "*wherein multiple work units are processed in parallel by multiple compute nodes*" is clearly performed. Therefore, since the Wood reference performs all of the above features previously applied to the Motamed reference, the Motamed reference regarding these features is no longer needed.

Lastly, regarding the features of Salgado, the features of the Wood reference regarding the segments, considered as work units, modified by the queuing features of Salgado, the combination of the two references with McIntyre discloses the Applicant's invention. Both inventions involve the processing PDL information (see Salgado col. 9, ln 56- col. 10, ln 12). Since both are associated with PDL processing (same field of endeavor) then it is appropriate to combine the two references. The Applicant did not go into any detail as to how the Salgado reference does not disclose the queue limitations in the claim language, or what specifically regarding the claim language is not met. The Examiner clearly believes that the combination of the Wood reference with the Salgado reference meet the claim limitations.

Therefore, the rejection in view of McIntyre, Wood and Salgado on the independent claims is maintained.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. Claims 1-4 and 9 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The non-statutory since the claim language mentions a method of performing functions that are not tied to a machine or apparatus. It is suggested that the claim limitations in the body of the claim mention the structure that is associated with a certain function, such as the parsing function or the queuing function. The dependent claims are also rejected because of their dependence on a rejected independent claim.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-4, 9-13, 18-21 and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over McIntyre '478 (USP 6690478) in view of Wood '934 (US Pub No 2004/0243934) and Salgado '621 (USP 6504621).

Re claim 1: McIntyre '478 discloses a method and apparatus for utilizing multiple versions of a page descriptor language comprising the steps of:

processing each of the plurality of work units by at least one compute node to convert each work unit into a second format (i.e. in McIntyre '478, a plurality of print jobs, considered as work units can be processed. The printer driver (114), considered

as the compute node, processes the incoming print jobs by recognizing the type of PDL is input into the system. The printer driver selects a PDL type from a PDL registry (112) to correspond with the incoming data and this PDL type chosen is used to convert the data into a different format, analogous to the second format, which is a low-level data stream; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

However, McIntyre '478 fails to teach parsing the datastream into a plurality of work units in a first format wherein each work unit may be processed independent of all other work units, wherein the plurality of work units are parsed from a single job, wherein each work unit may either be a data work unit or a control work unit, queueing each data work unit on a queue accessible by a plurality of compute nodes, wherein the processing of each work unit is independent of processing of the other work units and wherein multiple work units are processed in parallel by multiple compute nodes.

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses parsing the datastream into a plurality of work units in a first format (i.e. the inventions of McIntyre and Wood are both involved with processing PDL for an output in the system (same filed of endeavor). However, the apparatus and method of Wood '934 divides the PDL data stream to provide a plurality of PDL segments and then to create associated first and second data segments for each of the PDL segments. These segments are in a PDL format, analogous to a first format and the function of parsing is performed by the dividing and interpreting the PDL data stream into a plurality

of segments, which the plurality of segments are considered as the plurality of work units; see figs. 1-4; paragraphs [0004]-[0014] and [0025]-[0029])

wherein each work unit may be processed independent of all other work units (i.e. since the multiple segments are to be converted into another intermediate data stream, or another format, the segments can be considered analogous to a work unit. Wood also noted that the prior art divided the PDL data stream into independent data stream portions by multiple sub-processes. The prior art system ensured that the independence of the data stream segments were to be maintained through having translation information containing the appropriate "translation state" for each segment. Here, the segments could be processed by only depending on the system being able to process the translation information contained in the segment and not in any other segment (see paragraphs [0004]-[0006]). A further indicator of their independence was the various portions of the intermediate data stream portion being combined into a single intermediate data stream portion. Since these portions were not generated in chronological order, the portions would be generated at whatever rate the PDL interpreters could generate the information and then the system would have to then rearrange the various data in a correct chronological order to obtain the original data stream. This is an example of the portions being processed independently of each other (see paragraph [0007])),

wherein the plurality of work units are parsed from a single job (i.e. in the background, it was duly noted by Wood of the present inventions used to take a PDL datastream and parse it into multiple segments (mentioned in paragraph [0004]). Since

the multiple segments are to be converted into another intermediate data stream, or another format, the segments can be considered analogous to a work unit. This same feature of parsing a plurality of work units, or segments, from a single job is present in the overall invention of Wood; see paragraphs [0025]-[0027]),

wherein each work unit may either be a data work unit or a control work unit (i.e. the feature of the data work units, or data files, stored in the memory buffers (34 or 36) in order for these files to be accessed by the processors (40) to be processed for printing. The processors are considered analogous to the compute nodes. In reference to the control work unit, the segment data file (22) is used to include all the PDL commands necessary to interpret the segment by the PDL processors, and thus performs the feature of the control work unit; see paragraphs [0029]-[0033]);

queueing each data work unit on a queue accessible by a plurality of compute nodes (i.e. on the buffers, the files (20 and 22) contain information that enable the PDL information to be converted to the file the PDL information represents by the PDL processors; see paragraphs [0029]-[0033]) and

a control work unit processed by a computer node (i.e. in the system, the files (20 and 22) can be considered as either a data unit or a control unit. Since some files contain the instructions for the PDL processors, these files can be considered as control work units. These work units can be processed in the PDL processors considered as the computer nodes; see paragraphs [0029]-[0033]),

wherein multiple work units are processed in parallel by multiple compute nodes (i.e. the different PDL processors are considered analogous to compute nodes and these processors operate in parallel; see paragraphs [0008]-[0014] and [0031]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to parsing the datastream into a plurality of work units in a first format wherein each work unit may be processed independent of all other work units, wherein the plurality of work units are parsed from a single job, wherein each work unit may either be a data work unit or a control work unit, queueing each data work unit on a queue accessible by a plurality of compute nodes, wherein the processing of each work unit is independent of processing of the other work units and wherein multiple work units are processed in parallel by multiple compute nodes, incorporated in the device of McIntyre, in order to divide the PDL data stream to provide a plurality of PDL segments (as stated in Wood '934 paragraph [0014]).

However, the combination of McIntyre '478 and Wood '934 and fails to teach wherein each control unit may be an immediate control work unit or a scheduled work unit or an interrupt control work unit; queuing a scheduled control work unit at a tail of the queue to be processed by a compute node after all other work units presently in the queue; queuing an immediate control work unit at a head of the queue to be processed by a compute node before all other work units in the queue; forwarding an interrupt control work unit to a compute node immediately regardless of any work units in the queue.

However, this is well known in the art as evidenced by Salgado '621. Salgado '621 discloses wherein each control unit may be an immediate control work unit or a scheduled work unit or an interrupt control work unit (i.e. like to previously applied references, the reference of Salgado is used to process PDL information for output (same field of endeavor). However, with the different types of jobs in PDL being queued in Example 1, each type of job reflects a different type of control work unit. For example, the interrupt control work unit is analogous to the Authorized User Job in EXAMPLE 1 in column 19. Once this AU job reaches the system, it is immediately processed and interrupts the job that is currently being processed. The net print job is considered analogous to the immediate control work unit since this job is immediately placed at the head of the queue before all work units in the queue. This job is also similar to the copy job listed in the same column 18. The system report error job is considered as the schedule control work unit since it is paced at the end of the queue after all the other jobs presently in the queue; see col. 18, line 40 – col. 20 36);

queueing a scheduled control work unit at a tail of the queue to be processed after all other work units presently in the queue (i.e. when looking at EXAMPLE 1 in column 18, this example shows examples of the different types of control work units. The system error report job is considered as the schedule work unit, placed at the end of the queue and not processed until the jobs before the system error report job are processed beforehand; see col. 18, line 40 – col. 20, line 36);

queueing an immediate control work unit at a head of the queue to be processed before all other work units in the queue (i.e. the net print job is considered analogous to

the immediate control work unit since this job is immediately placed at the head of the queue before all work units in the queue. This job is also similar to the copy job listed in the same column 18; see col. 18, line 40 – col. 20, line 36);

forwarding an interrupt control work unit immediately regardless of any work units in the queue (i.e. the interrupt control work unit is analogous to the Authorized User Job in EXAMPLE 1 in column 19. Once this AU job reaches the system, it is immediately processed and interrupts the job that is currently being processed; see col. 18, line 40 – col. 20, line 36).

Therefore, in view of Salgado '621, it would have been obvious to one of ordinary skill at the time the invention was made to have the feature of wherein each control unit may be an immediate control work unit or a scheduled work unit or an interrupt control work unit; queuing a scheduled control work unit at a tail of the queue to be processed after all other work units presently in the queue; queuing an immediate control work unit at a head of the queue to be processed before all other work units in the queue; forwarding an interrupt control work unit immediately regardless of any work units in the queue incorporated in the device of McIntyre '478, as combined with the features of Wood '934, in order to have a system with a queue for structuring an order in which a plurality of print jobs is to be processed (as stated in Salgado col. 5, lines 6-29).

Re claim 2: The teachings of McIntyre '478 in view of Wood '934 and Salgado '621 are disclosed above.

McIntyre '478 discloses the method, wherein the parsing step (a) includes:

providing a plurality of sources, wherein each source is associated with at least one transform (i.e. in McIntyre '478, a method for registration and selection of multiple page description languages (i.e. personalities) is presented. The personalities, analogous to a plurality of sources, are associated with a transform, or conversion, in order to convert the input instructions into a printer dependent data stream interpreted by the printing subsystem (108) to produce an output page; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25);

instantiating at least one source of the plurality of sources, wherein the at least one instantiated source is associated with the datastream format (i.e. the printer driver (114) recognizes a realization of the personality related to the transform, or instantiates one transform, in order to perform a conversion, that is associated with received instructions. The printer driver (114) analyzes these received instructions and chooses one of the multiple PDLs registered in the PDL registry (112); see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25); and

utilizing the at least one source (i.e. once the printer driver (114) finds an appropriate version of a PDL registered within the PDL registry (112), the printer driver invokes the personality, or utilizes the personality analogous to the source, to convert the instructions from a high level language to a printer dependent data stream; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

However, McIntyre '478 fails to teach to parse the datastream.

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses to parse the datastream (i.e. the apparatus and method of Wood '934 divides

the PDL data stream to provide a plurality of PDL segments and then to create associated first and second data segments for each of the PDL segments. These segments are in a PDL format, analogous to a first format and the function of parsing is performed by the dividing and interpreting the PDL data stream into a plurality of segments, which the plurality of segments are considered as the plurality of work units; see figs. 1-4; paragraphs [0004]-[0014] and [0025]-[0029]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to parse the datastream in order to divide the PDL data stream to provide a plurality of PDL segments (as stated in Wood '934 paragraph [0014]).

Re claim 3: The teachings of McIntyre '478 in view of Wood '934 and Salgado '621 are disclosed above.

McIntyre '478 discloses the method, wherein the processing step (b) includes:

loading the at least one transform associated with the at least one instantiated source in the at least one compute node (i.e. the printer driver (114) recognizes a transform, or conversion, associated with at least one realization of the personality, analogous to an instantiated source, that is able to perform the transform and invokes the personality, or PDL, to perform the conversion of the received instructions. The action of invoking the personality after associated the PDL with the received data is analogous to immediately loading the transform in order to be utilized for transformation

of the received data; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25); and

utilizing the at least one transform to convert a work unit of the plurality of work units from the first format to the second format (i.e. in the system of McIntyre '478, the system is able to process a plurality of print jobs, considered as work units. The PDL utilized by the printer driver (114) to convert the received instructions, or print job, from a high level language to a printer dependent data, or language, is analogous to converting a work unit from a first format to a second format; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

Re claim 4: The teachings of McIntyre '478 in view of Wood '934 and Salgado '621 are disclosed above.

However, McIntyre '478 fails to teach the method further comprising: (c) load balancing the plurality of work units.

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses the method further comprising: (c) load balancing the plurality of work units (i.e. by sequentially assigning data files in Wood '934 to the next available PDL processor, the workload of processing data files (20) and (22) may be automatically load-balanced between various PDL processors. The data files are analogous to the plurality of work units; see paragraph [0033]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to have the step of load balancing the plurality

of work units in order to automatically load-balance the workload of processing data files in the system (as stated in Wood '934 paragraph [0033]).

Re claim 9: The teachings of McIntyre '478 in view of Wood '934 and Salgado '621 are disclosed above.

McIntyre '478 discloses the method, wherein the at least one source is instantiated as a dynamic library (i.e. when using a transform to convert incoming data into another form, the printer driver (114) requests for a transform to convert from the incoming PDL into a low-level language for the printer to understand. This is performed by linking the incoming data to the specific personality that will perform the transformation of the data stream to the low-level format. This process is similar to a library with a collection of subprograms used to develop other pieces of information used by the system and provide the function of transformation that is linked to a certain input language in the invention of McIntyre '478; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

Re claim 10: McIntyre '478 discloses a method and apparatus for utilizing multiple versions of a page descriptor language, the program instructions for:

processing each of the plurality of work units by at least one compute node to convert each data work unit into a second format (i.e. in McIntyre '478, a plurality of print jobs, considered as work units can be processed. The printer driver (114), considered as the compute node, processes the incoming print jobs by recognizing the type of PDL

is input into the system. The printer driver selects a PDL type from a PDL registry (112) to correspond with the incoming data and this PDL type chosen is used to convert the data into a different format, analogous to the second format, which is a low-level data stream. Also, with the above actions capable of being performed on a storage medium having stored the executable instructions to implement the teachings of the invention of McIntyre '478, the above feature of a computer readable medium containing program instructions is performed; see figs. 1, 3 and 6; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160, col. 6, lines 1-25 and col. 7, lines 7-23).

However, McIntyre '478 fails to teach parsing the datastream into a plurality of work units in a first format wherein each work unit may be processed independent of all other work units, wherein the plurality of work units are parsed from a single job, wherein each work unit may either be a data work unit or a control work unit, queueing each data work unit on a queue accessible by a plurality of compute nodes, wherein the processing of each work unit is independent of processing of the other work units and wherein multiple work units are processed in parallel by multiple compute nodes.

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses parsing the datastream into a plurality of work units in a first format (i.e. like the reference of McIntyre, the reference of Wood is used to process PDL information for output (same field of endeavor). However, the apparatus and method of Wood '934 divides the PDL data stream to provide a plurality of PDL segments and then to create associated first and second data segments for each of the PDL segments. These segments are in a PDL format, analogous to a first format and the function of parsing is

performed by the dividing and interpreting the PDL data stream into a plurality of segments, which the plurality of segments are considered as the plurality of work units; see figs. 1-4; paragraphs [0004]-[0014] and [0025]-[0029]),

wherein each work unit may be processed independent of all other work units (i.e. since the multiple segments are to be converted into another intermediate data stream, or another format, the segments can be considered analogous to a work unit. Wood also noted that the prior art divided the PDL data stream into independent data stream portions by multiple sub-processes. The prior art system ensured that the independence of the data stream segments were to be maintained through having translation information containing the appropriate "translation state" for each segment. Here, the segments could be processed by only depending on the system being able to process the translation information contained in the segment and not in any other segment (see paragraphs [0004]-[0006]). A further indicator of their independence was the various portions of the intermediate data stream portion being combined into a single intermediate data stream portion. Since these portions were not generated in chronological order, the portions would be generated at whatever rate the PDL interpreters could generate the information and then the system would have to then rearrange the various data in a correct chronological order to obtain the original data stream. This is an example of the portions being processed independently of each other (see paragraph [0007])),

wherein the plurality of work units are parsed from a single job (i.e. in the background, it was duly noted by Wood of the present inventions used to take a PDL

datastream and parse it into multiple segments (mentioned in paragraph [0004]). Since the multiple segments are to be converted into another intermediate data stream, or another format, the segments can be considered analogous to a work unit. This same feature of parsing a plurality of work units, or segments, from a single job is present in the overall invention of Wood; see paragraphs [0025]-[0027]),

wherein each work unit may either be a data work unit or a control work unit (i.e. the feature of the data work units, or data files, stored in the memory buffers (34 or 36) in order for these files to be accessed by the processors (40) to be processed for printing. The processors are considered analogous to the compute nodes. In reference to the control work unit, the segment data file (22) is used to include all the PDL commands necessary to interpret the segment by the PDL processors, and thus performs the feature of the control work unit; see paragraphs [0029]-[0033]);

queueing each data work unit on a queue accessible by a plurality of compute nodes (i.e. on the buffers, the files (20 and 22) contain information that enable the PDL information to be converted to the file the PDL information represents by the PDL processors; see paragraphs [0029]-[0033]) and

a control work unit processed by a computer node (i.e. in the system, the files (20 and 22) can be considered as either a data unit or a control unit. Since some files contain the instructions for the PDL processors, these files can be considered as control work units. These work units can be processed in the PDL processors considered as the computer nodes; see paragraphs [0029]-[0033]),

wherein multiple work units are processed in parallel by multiple compute nodes (i.e. the different PDL processors are considered analogous to compute nodes and these processors operate in parallel; see paragraphs [0008]-[0014] and [0031]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to parsing the datastream into a plurality of work units in a first format wherein each work unit may be processed independent of all other work units, wherein the plurality of work units are parsed from a single job, wherein each work unit may either be a data work unit or a control work unit, queueing each data work unit on a queue accessible by a plurality of compute nodes, wherein the processing of each work unit is independent of processing of the other work units and wherein multiple work units are processed in parallel by multiple compute nodes in order to divide the PDL data stream to provide a plurality of PDL segments (as stated in Wood '934 paragraph [0014]).

However, the combination of McIntyre '478 and Wood '934 fails to teach wherein each control unit may be an immediate control work unit or a scheduled work unit or an interrupt control work unit; queuing a scheduled control work unit at a tail of the queue to be processed after all other work units presently in the queue; queuing an immediate control work unit at a head of the queue to be processed before all other work units in the queue; forwarding an interrupt control work unit immediately regardless of any work units in the queue.

However, this is well known in the art as evidenced by Salgado '621. Salgado '621 discloses wherein each control unit may be an immediate control work unit or a

scheduled work unit or an interrupt control work unit (i.e. like the above applied references, the reference of Salgado is used to process PDL information for output (same field of endeavor). However, with the different types of jobs being queued in Example 1, each type of job reflects a different type of control work unit. For example, the interrupt control work unit is analogous to the Authorized User Job in EXAMPLE 1 in column 19. Once this AU job reaches the system, it is immediately processed and interrupts the job that is currently being processed. The net print job is considered analogous to the immediate control work unit since this job is immediately placed at the head of the queue before all work units in the queue. This job is also similar to the copy job listed in the same column 18. The system report error job is considered as the schedule control work unit since it is paced at the end of the queue after all the other jobs presently in the queue; see col. 18, line 40 – col. 20 36);

queueing a scheduled control work unit at a tail of the queue to be processed after all other work units presently in the queue (i.e. when looking at EXAMPLE 1 in column 18, this example shows examples of the different types of control work units. The system error report job is considered as the schedule work unit, placed at the end of the queue and not processed until the jobs before the system error report job are processed beforehand; see col. 18, line 40 – col. 20, line 36);

queueing an immediate control work unit at a head of the queue to be processed before all other work units in the queue (i.e. the net print job is considered analogous to the immediate control work unit since this job is immediately placed at the head of the

queue before all work units in the queue. This job is also similar to the copy job listed in the same column 18; see col. 18, line 40 – col. 20, line 36);

forwarding an interrupt control work unit immediately regardless of any work units in the queue (i.e. the interrupt control work unit is analogous to the Authorized User Job in EXAMPLE 1 in column 19. Once this AU job reaches the system, it is immediately processed and interrupts the job that is currently being processed; see col. 18, line 40 – col. 20, line 36).

Therefore, in view of Salgado '621, it would have been obvious to one of ordinary skill at the time the invention was made to have the feature of wherein each control unit may be an immediate control work unit or a scheduled work unit or an interrupt control work unit; queuing a scheduled control work unit at a tail of the queue to be processed after all other work units presently in the queue; queuing an immediate control work unit at a head of the queue to be processed before all other work units in the queue; forwarding an interrupt control work unit immediately regardless of any work units in the queue incorporated in the device of McIntyre '478, as combined with the features of Wood '934, in order to have a system with a queue for structuring an order in which a plurality of print jobs is to be processed (as stated in Salgado col. 5, lines 6-29).

Re claim 11: The teachings of McIntyre '478 in view of Wood '934 and Salgado '621 are disclosed above.

McIntyre '478 discloses the computer readable medium of claim 10, wherein the parsing instruction (a) includes:

providing a plurality of sources, wherein each source is associated with at least one transform (i.e. in McIntyre '478, a method for registration and selection of multiple page description languages (i.e. personalities) is presented. The personalities, analogous to a plurality of sources, are associated with a transform, or conversion, in order to convert the input instructions into a printer dependent data stream interpreted by the printing subsystem (108) to produce an output page; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25);

instantiating at least one source of the plurality of sources, wherein the at least one instantiated source is associated with the datastream format (i.e. the printer driver (114) recognizes a realization of the personality related to the transform, or instantiates one transform, in order to perform a conversion, that is associated with received instructions. The printer driver (114) analyzes these received instructions and chooses one of the multiple PDLs registered in the PDL registry (112); see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25); and

utilizing the at least one source (i.e. once the printer driver (114) finds an appropriate version of a PDL registered within the PDL registry (112), the printer driver invokes the personality, or utilizes the personality analogous to the source, to convert the instructions from a high level language to a printer dependent data stream; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

However, McIntyre '478 fails to teach to parse the datastream.

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses to parse the datastream (i.e. the apparatus and method of Wood '934 divides

the PDL data stream to provide a plurality of PDL segments and then to create associated first and second data segments for each of the PDL segments. These segments are in a PDL format, analogous to a first format and the function of parsing is performed by the dividing and interpreting the PDL data stream into a plurality of segments, which the plurality of segments are considered as the plurality of work units; see figs. 1-4; paragraphs [0004]-[0014] and [0025]-[0029]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to parse the datastream in order to divide the PDL data stream to provide a plurality of PDL segments (as stated in Wood '934 paragraph [0014]).

Re claim 12: The teachings of McIntyre '478 in view of Wood '934 and Salgado '621 are disclosed above.

McIntyre '478 discloses the computer readable medium of claim 11, wherein the processing instruction includes:

loading the at least one transform associated with the at least one instantiated source in the at least one compute node (i.e. the printer driver (114) recognizes a transform, or conversion, associated with at least one realization of the personality, analogous to an instantiated source, that is able to perform the transform and invokes the personality, or PDL, to perform the conversion of the received instructions. The action of invoking the personality after associated the PDL with the received data is analogous to immediately loading the transform in order to be utilized for transformation

of the received data; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25); and

utilizing the at least one transform to convert a work unit of the plurality of work units from the first format to the second format (i.e. in the system of McIntyre '478, the system is able to process a plurality of print jobs, considered as work units. The PDL utilized by the printer driver (114) to convert the received instructions, or print job, from a high level language to a printer dependent data, or language, is analogous to converting a work unit from a first format to a second format; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

Re claim 13: The teachings of McIntyre '478 in view of Wood '934 and Salgado '621 are disclosed above.

However, McIntyre '478 fails to teach the computer readable medium further comprising: load balancing the plurality of work units.

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses the computer readable medium further comprising: (c) load balancing the plurality of work units (i.e. by sequentially assigning data files in Wood '934 to the next available PDL processor, the workload of processing data files (20) and (22) may be automatically load-balanced between various PDL processors. The data files are analogous to the plurality of work units; see paragraph [0033]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to have the step of load balancing the plurality

of work units in order to automatically load-balance the workload of processing data files in the system (as stated in Wood '934 paragraph [0033]).

Re claim 18: The teachings of McIntyre '478 in view of Wood '934 and Salgado '621 are disclosed above.

McIntyre '478 discloses the computer readable medium, wherein the at least one source is instantiated as a dynamic library (i.e. when using a transform to convert incoming data into another form, the printer driver (114) requests for a transform to convert from the incoming PDL into a low-level language for the printer to understand. This is performed by linking the incoming data to the specific personality that will perform the transformation of the data stream to the low-level format. This process is similar to a library with a collection of subprograms used to develop other pieces of information used by the system and provide the function of transformation that is linked to a certain input language in the invention of McIntyre '478; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

Re claim 19: McIntyre '478 discloses a method and apparatus for utilizing multiple versions of a page descriptor language comprising:

a central component for receiving the datastream in a first format (i.e. McIntyre '478 discloses a control driver that receives the datastream in a high-level language, or a first format; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25);

a plurality of sources in the central component, wherein each of the plurality of sources is associated with at least one transform (i.e. the plurality of personalities, considered as sources, are managed by both the control driver (104) and the boot agent (102). Since the control driver manages the personalities, this can be considered as having the personalities in the control driver (104) to be managed; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25); and

at least one compute node coupled to the central component (i.e. the printer driver (114), considered as the compute node, is coupled to the control driver (104); see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25),

wherein the central component instantiates at least one source of the plurality of sources (i.e. the control driver (104) uses the printer driver (114) to create a particular realization of a printer description language, or instantiates, through recognizing the personality in the system, which is analogous to the plurality of sources; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25), and distributes each of the work units to the at least one compute node,

wherein the at least one compute node converts each data work unit into a second format (i.e. in McIntyre '478, a plurality of print jobs, considered as work units can be processed. The printer driver (114), considered as the compute node, processes the incoming print jobs by recognizing the type of PDL is input into the system. The printer driver selects a PDL type from a PDL registry (112) to correspond with the incoming data and this PDL type chosen is used to convert the data into a

different format, analogous to the second format, which is a low-level data stream; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

However, McIntyre '478 fails to teach a queue, at least one compute node coupled to the central component via the queue, parses the datastream into a plurality of work units in the first format, wherein each work unit may be processed independent of all other work units, wherein the plurality of work units are parsed from a single job, wherein the central component distributes each of the work units to the at least one compute node by queueing each data work unit on a queue accessible by a plurality of compute nodes and a control work unit to be processed by a computer node, wherein the at least one compute node converts each data work unit into a second format independent of all other compute nodes operable on other work units, and wherein at least two compute nodes are operable in parallel to convert at least two data work units in parallel.

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses a queue (i.e. like the McIntyre reference, the Wood reference processes PDL information for output (same field of endeavor). However, in the system, the memory buffers (34 and 36) can be considered as the queue or the memory buffer (40) can be considered as the queue; see paragraphs [0025]-[0029]); and

at least one compute node coupled to the central component via the queue (i.e. in the system, a PDL processor is coupled to the schedule segments portion (24),

considered as the central component, via the processor FIFO, considered as the queue in this scenario; see paragraphs [0025]-[0029]),

parses the datastream into a plurality of work units in the first format (i.e. the apparatus and method of Wood '934 divides the PDL data stream to provide a plurality of PDL segments and then to create associated first and second data segments for each of the PDL segments. These segments are in a PDL format, analogous to a first format and the function of parsing is performed by the dividing and interpreting the PDL data stream into a plurality of segments, which the plurality of segments are considered as the plurality of work units; see figs. 1-4; paragraphs [0004]-[0014] and [0025]-[0029]),

wherein each work unit may be processed independent of all other work units (i.e. since the multiple segments are to be converted into another intermediate data stream, or another format, the segments can be considered analogous to a work unit. Wood also noted that the prior art divided the PDL data stream into independent data stream portions by multiple sub-processes. The prior art system ensured that the independence of the data stream segments were to be maintained through having translation information containing the appropriate "translation state" for each segment. Here, the segments could be processed by only depending on the system being able to process the translation information contained in the segment and not in any other segment (see paragraphs [0004]-[0006]). A further indicator of their independence was the various portions of the intermediate data stream portion being combined into a single intermediate data stream portion. Since these portions were not generated in chronological order, the portions would be generated at whatever rate the PDL

interpreters could generate the information and then the system would have to then rearrange the various data in a correct chronological order to obtain the original data stream. This is an example of the portions being processed independently of each other (see paragraph [0007])),

wherein the plurality of work units are parsed from a single job (i.e. in the background, it was duly noted by Wood of the present inventions used to take a PDL datastream and parse it into multiple segments (mentioned in paragraph [0004]). Since the multiple segments are to be converted into another intermediate data stream, or another format, the segments can be considered analogous to a work unit. This same feature of parsing a plurality of work units, or segments, from a single job is present in the overall invention of Wood; see paragraphs [0025]-[0027]),

wherein the central component distributes each of the work units to the at least one compute node by queueing each data work unit on a queue accessible by a plurality of compute nodes (i.e. in the system, the schedule segments portion is used as the central component to distribute global and data segment files, considered as work units, to the plurality of PDL processors by queueing the global and data segment files in the memory buffers (40), which are accessible from the PDL processor, considered analogous to the computer nodes; see fig. 3; see paragraphs [0025]-[0029]), and

a control work unit to be processed by a computer node (i.e. in the system, the files (20 and 22) can be considered as either a data unit or a control unit. Since some files contain the instructions for the PDL processors, these files can be considered as

control work units. These work units can be processed in the PDL processors considered as the computer nodes; see paragraphs [0029]-[0033]),

wherein the at least one computer node converts each data work unit into a second format independent of all other compute nodes operable on other work units (i.e. in the system, the PDL processors are used to convert a segment of a job. The PDL processors shown in figure 5B are used to convert segment information into another format and these separate segments are produced by the PDL processor 1 that is independent from the PDL processor 2. The segments that are produced on PDL processor 1 are also independent from the segments on PDL processor 2; see paragraphs [0031]-[0037]) and

wherein at least two compute nodes are operable in parallel to convert at least two data work units in parallel (i.e. the different PDL processors are considered analogous to compute nodes and these processors operate in parallel when processing different segments of a job; see paragraphs [0008]-[0014] and [0031]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to a queue, at least one compute node coupled to the central component via the queue, parses the datastream into a plurality of work units in the first format, wherein each work unit may be processed independent of all other work units, wherein the plurality of work units are parsed from a single job, wherein the central component distributes each of the work units to the at least one compute node by queuing each data work unit on a queue accessible by a plurality of compute nodes and a control work unit to be processed by a computer node, wherein

the at least one compute node converts each data work unit into a second format independent of all other compute nodes operable on other work units, and wherein at least two compute nodes are operable in parallel to convert at least two data work units in parallel in order to divide the PDL data stream to provide a plurality of PDL segments (as stated in Wood '934 paragraph [0014]).

However, the combination of McIntyre '478 and Wood '934 fails to teach the features of by queuing a scheduled control work unit at a tail of the queue to be processed after all other work units presently in the queue, by queuing an immediate control work unit at a head of the queue to be processed before all other work units in the queue and by forwarding an interrupt control work unit immediately regardless of any work units in the queue.

However, this is well known in the art as evidenced by Salgado '621. Salgado '621 discloses the feature of by queueing a scheduled control work unit at a tail of the queue to be processed after all other work units presently in the queue (i.e. like the above applied references, the reference of Salgado is used to process PDL information for output (same field of endeavor). However, when looking at EXAMPLE 1 in column 18, this example shows examples of the different types of control work units. The system error report job is considered as the schedule work unit, placed at the end of the queue and not processed until the jobs before the system error report job are processed beforehand; see col. 18, line 40 – col. 20, line 36);

by queueing an immediate control work unit at a head of the queue to be processed before all other work units in the queue (i.e. The net print job is considered

analogous to the immediate control work unit since this job is immediately placed at the head of the queue before all work units in the queue. This job is also similar to the copy job listed in the same column 18; see col. 18, line 40 – col. 20, line 36);

by forwarding an interrupt control work unit immediately regardless of any work units in the queue (i.e. the interrupt control work unit is analogous to the Authorized User Job in EXAMPLE 1 in column 19. Once this AU job reaches the system, it is immediately processed and interrupts the job that is currently being processed; see col. 18, line 40 – col. 20, line 36).

Therefore, in view of Salgado '621, it would have been obvious to one of ordinary skill at the time the invention was made to have the features of the features of by queuing a scheduled control work unit at a tail of the queue to be processed after all other work units presently in the queue, by queuing an immediate control work unit at a head of the queue to be processed before all other work units in the queue and by forwarding an interrupt control work unit immediately regardless of any work units in the queue incorporated in the device of McIntyre '478, as combined with the features of Wood '934, in order to have a system with a queue for structuring an order in which a plurality of print jobs is to be processed (as stated in Salgado col. 5, lines 6-29).

Re claim 20: The teachings of McIntyre '478 in view of Wood '934 and Salgado '621 are disclosed above.

McIntyre '478 discloses the system of claim 19, wherein each of the at least one compute nodes loads the at least one transform as a dynamic library (i.e. the printer

driver (114), considered as the compute node, is able to utilize a personality, analogous to a transform, in interpreting incoming data. The incoming data is analyzed by the control driver (104) and the link between the incoming data and the appropriate personality to use for interpretation is made. The link of the incoming data to a collection of software used to change or provide services to other programs is an example of a dynamic library; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25) and utilizes the at least one transforms to convert a work unit in the first format to the second format (i.e. the printer driver (114) utilizes one of the personalities, or transforms, to convert a print job, considered as a work unit, from a high-level language to a low-level language that the printer can understand. This is analogous to converting from a first to a second format; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

Re claim 21: The teachings of McIntyre '478 in view of Wood '934 and Salgado '621 are disclosed above.

However, McIntyre '478 fails to teach the system of claim 19, wherein the central component further includes: a load balancing mechanism coupled to the at least one source for distributing the plurality of work units to the at least one compute node.

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses a load balancing mechanism coupled to the at least one source for distributing the plurality of work units to the at least one compute node (i.e. in Wood '934, the segment data files (22) are in a buffer memory (40) with a scheduler process (24),

analogous to a load balancing mechanism, that accesses the data files and distributes the data files, analogous to work units, to the PDL processors, which are analogous to the compute nodes; see figs. 1 and 4; paragraphs [0030]-[0034]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to have a load balancing mechanism coupled to the source for distributing a plurality of work units to the compute node in an order dependent manner (as stated in Wood '934 paragraphs [0030]-[0034]).

Re claim 25: The teachings of McIntyre '478 in view of Wood '934 and Salgado '621 are disclosed above.

McIntyre '478 discloses the system, wherein the at least one source is instantiated as a dynamic library (i.e. when using a transform to convert incoming data into another form, the printer driver (114) requests for a transform to convert from the incoming PDL into a low-level language for the printer to understand. This is performed by linking the incoming data to the specific personality that will perform the transformation of the data stream to the low-level format. This process is similar to a library with a collection of subprograms used to develop other pieces of information used by the system and provide the function of transformation that is linked to a certain input language in the invention of McIntyre '478; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

Conclusion

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

7. Vennekens '711 (USP 5652711) discloses a system that parses a data stream into a plurality of PDL data stream segments in a high-level language and each of these segments are stored in a FIFO queue before the segments are output to a sub-process where the segments are converted to a low-level language. Load balancing is performed in regards to the segments in the system.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to CHAD DICKERSON whose telephone number is (571)270-1351. The examiner can normally be reached on Mon. thru Thur. 9:00-6:30 Fri. 9:00-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Twyler Haskins can be reached on (571)-272-7406. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/C. D./
/Chad Dickerson/
Examiner, Art Unit 2625

/Mark K Zimmerman/
Supervisory Patent Examiner, Art Unit 2625